



MACHINE LEARNING FOR EVERYONE.

Avneesh Jain (CodeKraft) and
Sambuddha Roy (LinkedIn)

WHAT IS MACHINE LEARNING?



WHAT IS MACHINE LEARNING?

- Enable computers/machines to “learn” from existing (i.e. historical) data.
- What is the learning used for?
 - Predict – new data from old (for eg. classification)
 - Extract hidden structure (for eg. clustering)
 - Summarize data
 - ... many other use-cases

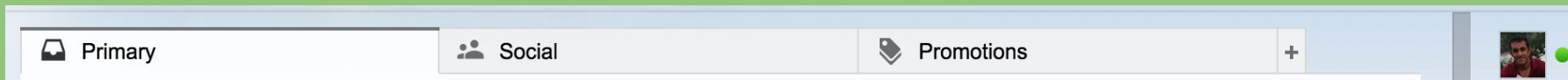
MACHINE LEARNING: CLASSIFICATION

According to wikipedia, “*classification* is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.”

MACHINE LEARNING: CLASSIFICATION

Examples abound:

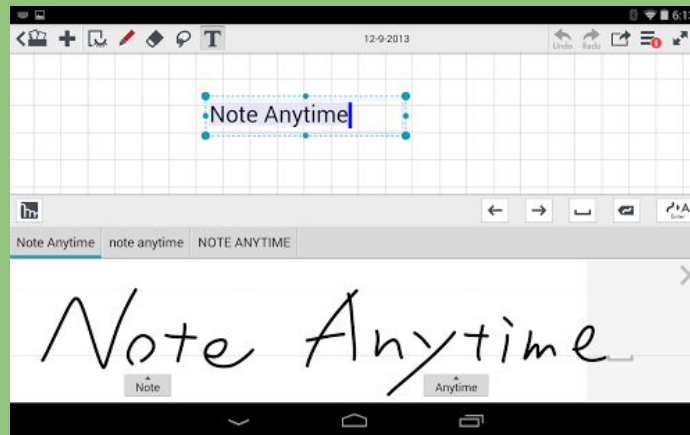
- Spam Classification
- GMail classifies your email as “Primary”, “Social” or “Promotions”



- “classification” of a newly bookmarked URL into the correct Bookmarks folder (was introduced as a feature for some time in Chrome)!

MACHINE LEARNING: CLASSIFICATION

- Handwriting Recognition.
- Speech Recognition



** Courtesy Google Images.

MACHINE LEARNING: CLASSIFICATION

- Cat or Dog?
- Car or Truck?



** Courtesy Google Images.

AND MANY OTHER APPLICATIONS...

- Cancer diagnosis
- Video classification
- Click Stream Analysis

MANY, MANY...

- Internet Traffic Interception,
- Sentiment Analysis,
- ...



** Courtesy Google Images.

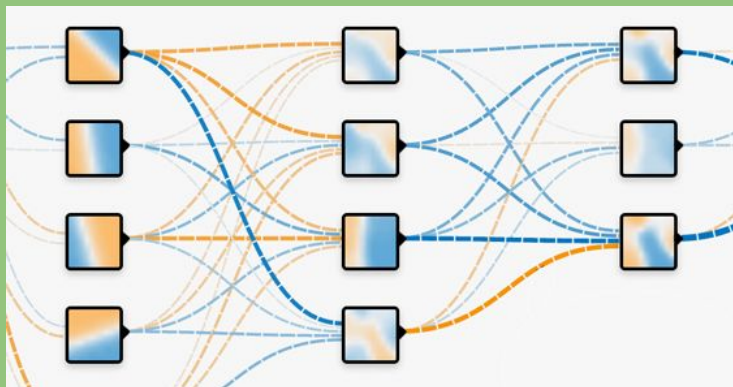
ROADMAP I.E. TOPICS WE WILL COVER

- Preliminaries for solving classification problems.
 - For this, we discuss features, decision boundaries,
 - Linear vs. non-linear classifiers.
- Discuss feature transformations
 - When a classifier is really linear, in some transformed features.
- Popular linear classifiers used in practice.
 - We discuss Naive Bayes in some detail.
 - Some details of logistic regression.
- Essential considerations:
 - Training, testing.
 - Metrics: AUC, Precision, Recall, F1-score, ...



ROADMAP I.E. TOPICS WE WILL COVER.

- Sentiment Analysis
 - Explain problem.
 - Describe dataset, separate into training and test datasets.
 - Train a Naive Bayes model to the problem.
- On to non-linear territory... Deep Learning.
 - Why is deep learning magical?



** Courtesy Google Images.

AND TOPICS THAT WE WON'T...

- Differences between
 - Supervised
 - Semi-supervised
 - Unsupervised
- Differences between
 - Regression
 - Classification
- Differences between
 - Discriminative
 - Generative
- Overfitting, regularization



AND TOPICS THAT WE WON'T...

- Bias-variance tradeoffs.
- Statistical Significance of parameters/weights
 - p-values etc.
- Cross-validation
 - Hold-out sets, etc.
- Correlated features.



AND TOPICS THAT WE WON'T...

- About deep networks:
 - Autoencoders, RBMs
 - RNNs, CNNs
 - GANs
 - Activation functions,
 - Dropout, etc.



CLASSIFICATION: HOW DO WE START?

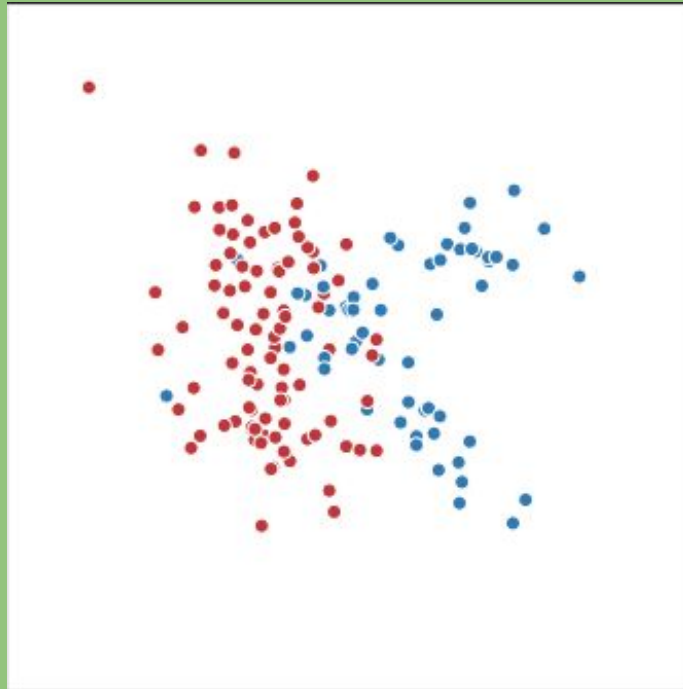
- Collect Features/Attributes!
- Eg. for the car vs. truck problem:
 - Number of wheels
 - Height of the vehicle
 - Length of the vehicle
 - Radius of wheels
 - Thickness of the wheels
 - What else?

CLASSIFICATION: HOW DO WE START?

- We can thereby “represent” an object (for instance a Buick) in “feature-space”.
- For instance, a Buick = (4, 1.5m, 3m, 0.5m, 0.2m...)

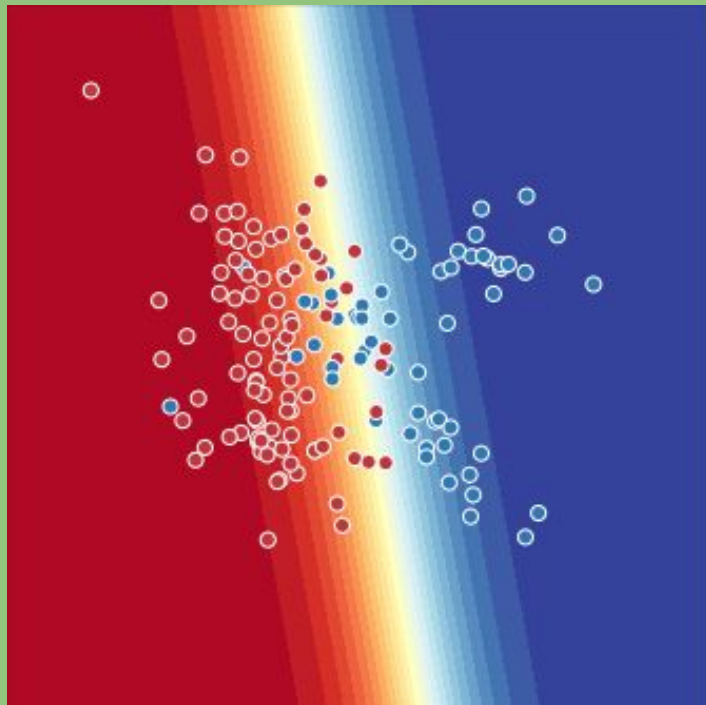
FEATURE SPACE

Visualize the data: We plot the data points (according to their feature vectors) in n-diml. space.



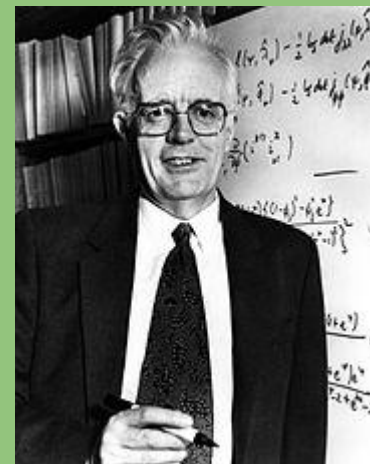
A LINEAR CLASSIFIER?

A linear classifier is one whose decision boundary is a line (or a hyperplane in feature space).



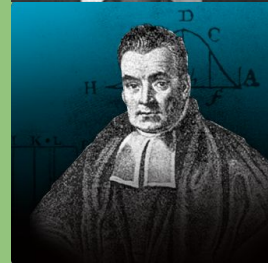
EXAMPLES OF LINEAR CLASSIFIERS

- Logistic Regression
 - Most popular, used widely in diverse areas such as
 - **ADVERTISING,**
 - **FINTECH,**
 - **MANY OTHERS.**



EXAMPLES OF LINEAR CLASSIFIERS

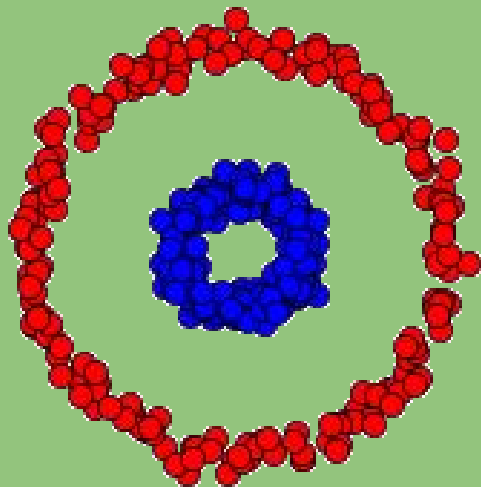
- Support Vector Machines
 - Sound theory backing, but heavier on optimization.
 - *IMAGE CLASSIFICATION*
 - *BIO-INFORMATICS*
- Naive Bayes.
 - Bayes'ed on Bayes' Theorem. Used often in
 - *SENTIMENT ANALYSIS ETC.*



DO LINEAR CLASSIFIERS ALWAYS SUFFICE?

NO!

- What if the two classes are *not* separable by a hyperplane?



DO LINEAR CLASSIFIERS ALWAYS SUFFICE?

NO!

- What if the two classes are *not* separable by a hyperplane?
- Some problems allow ingenious constructions by which we can escape non-linearity. For instance, if it turns out that by a transformation:

DO LINEAR CLASSIFIERS ALWAYS SUFFICE?

NO!

- What if the two classes are *not* separable by a hyperplane?
- Some problems allow ingenious constructions by which we can escape non-linearity. For instance, if it turns out that by a transformation:
 - $(x_1, x_2) \rightarrow (\log x_1, \log x_2)$, the classes become *linearly separable*!

DO LINEAR CLASSIFIERS ALWAYS SUFFICE?

NO!

- What if the two classes are *not* separable by a hyperplane?
- Some problems allow ingenious constructions by which we can escape non-linearity. For instance, if it turns out that by a transformation:
 - $(x_1, x_2) \rightarrow (\log x_1, \log x_2)$, the classes become *linearly separable*!
- How do we figure out the right **transformation**?

DO LINEAR CLASSIFIERS ALWAYS SUFFICE?

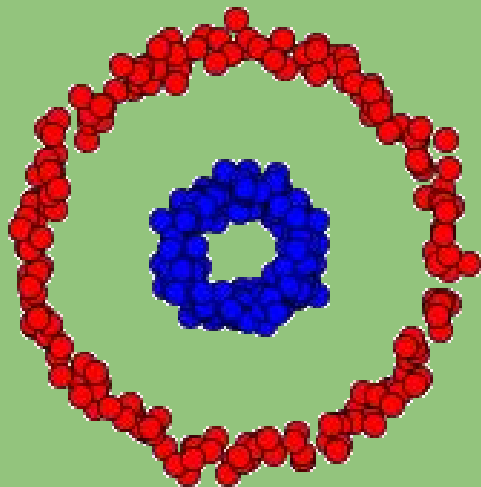
NO!

- What if the two classes are *not* separable by a hyperplane?
- Some problems allow ingenious constructions by which we can escape non-linearity. For instance, if it turns out that by a transformation:
 - $(x_1, x_2) \rightarrow (\log x_1, \log x_2)$, the classes become *linearly separable*!
- How do we figure out the right **transformation**?
 - Intelligent feature engineering
 - Rigorous experimentation, accompanied with evaluation of metrics, etc.
 - Smart guesswork

DO LINEAR CLASSIFIERS ALWAYS SUFFICE?

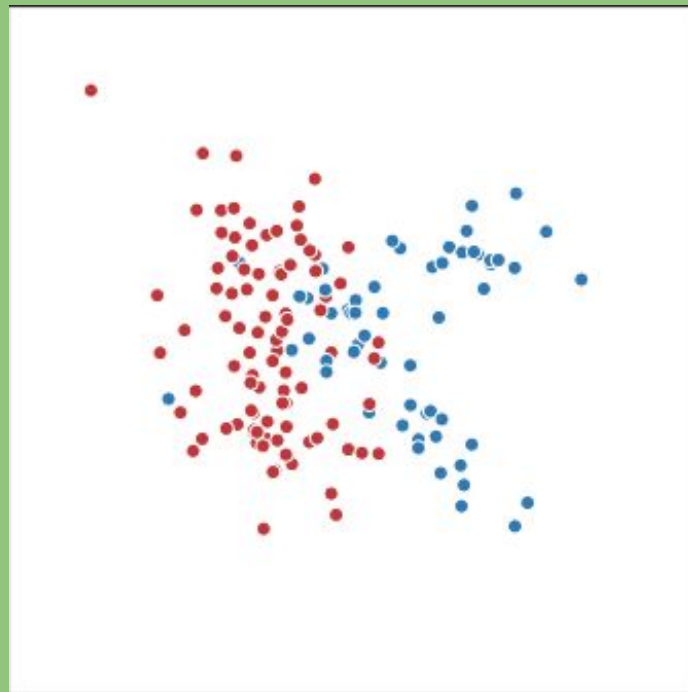
NO!

- What if the two classes are *not* separable by a hyperplane?
- We will come back to this in a moment.



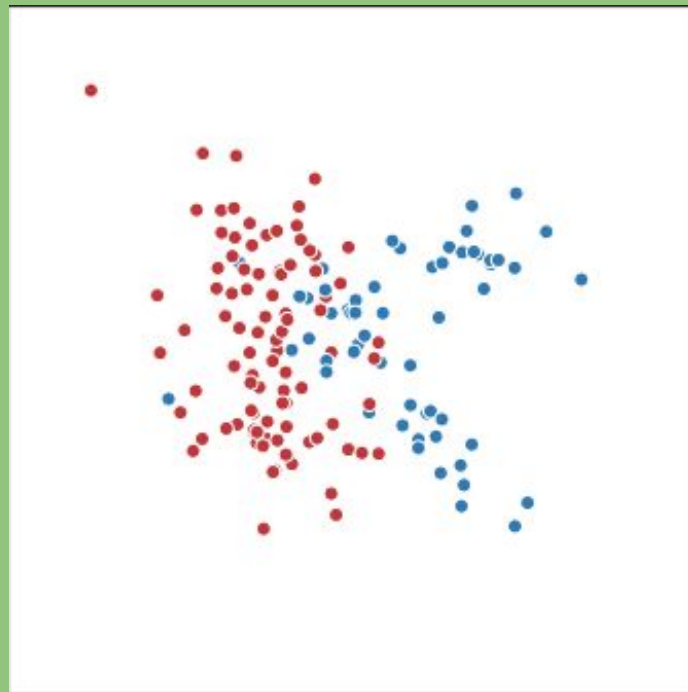
BACK TO LINEAR CLASSIFIERS

- What we we trying to learn?
 - The parameters determining the line.



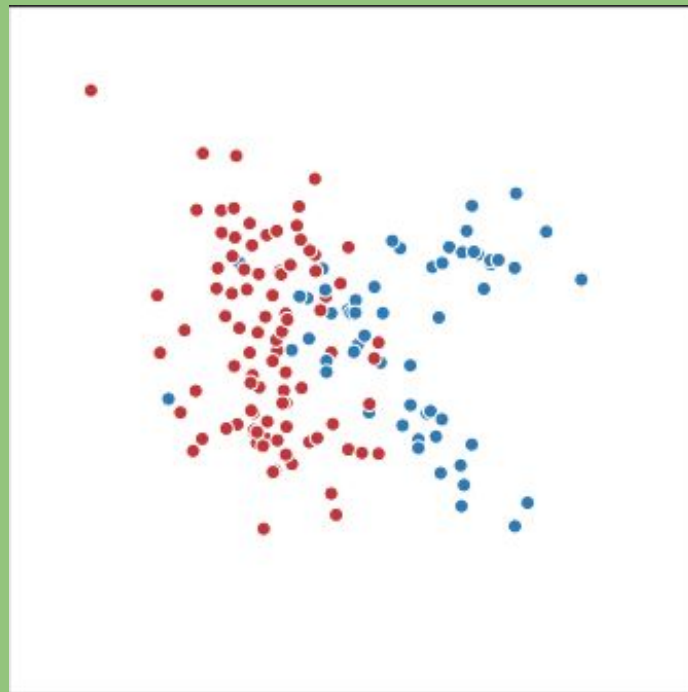
BACK TO LINEAR CLASSIFIERS

- What we we trying to learn?
 - The parameters determining the line.
- Which line? How do we make the choice?
 - Associate an objective function.



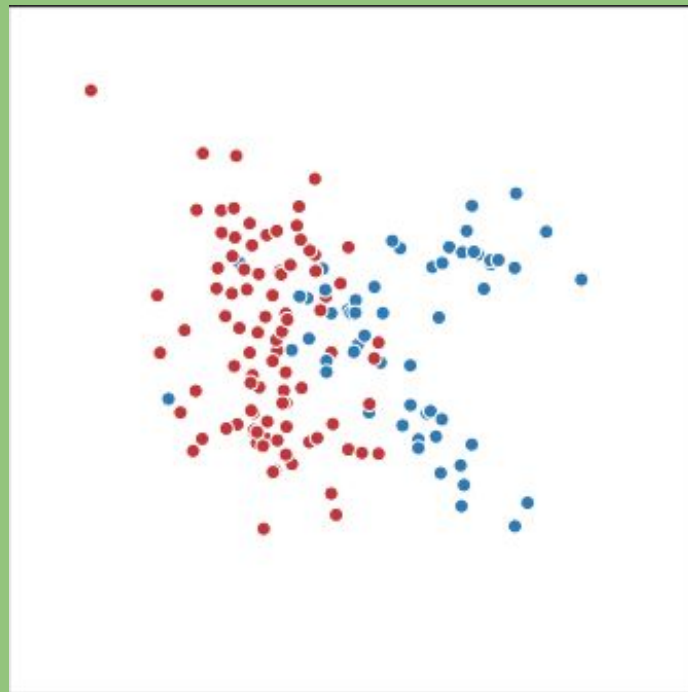
BACK TO LINEAR CLASSIFIERS

- What we we trying to learn?
 - The parameters determining the line.
- Which line? How do we make the choice?
 - Associate an objective function.
- However we are allowed to...
 - Transform features!



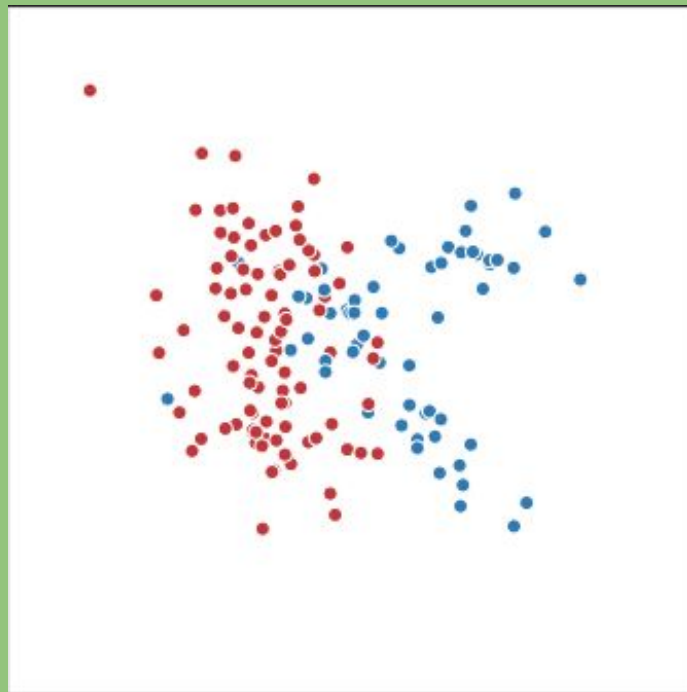
FEATURE TRANSFORMATION.

- Suppose we were to “transform” features



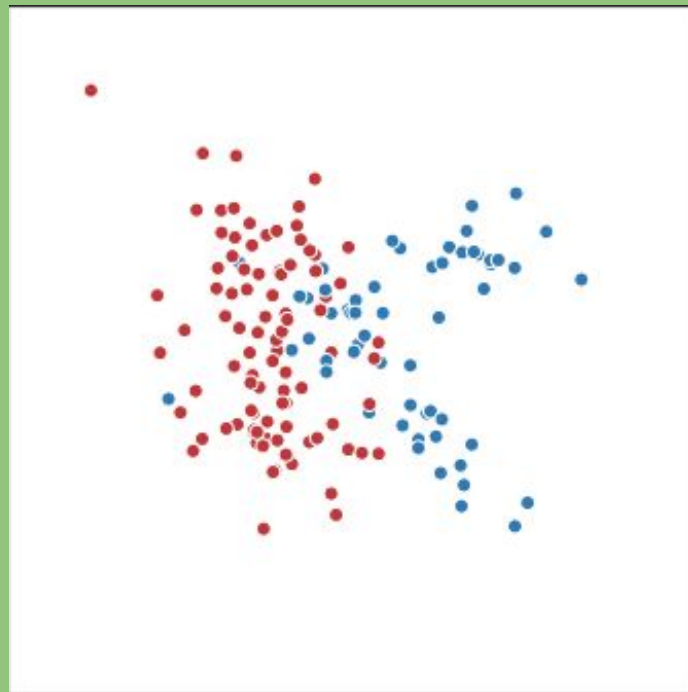
FEATURE TRANSFORMATION.

- Suppose we were to “transform” features x, y
 - $x \rightarrow 2x$ (i.e. $x_{\text{new}} = 2x_{\text{old}}$)
 - $x \rightarrow (x + y)$
 - I.e. linear transforms
 - Corresponds to “stretches” of featurespace
 - Linear stays linear.



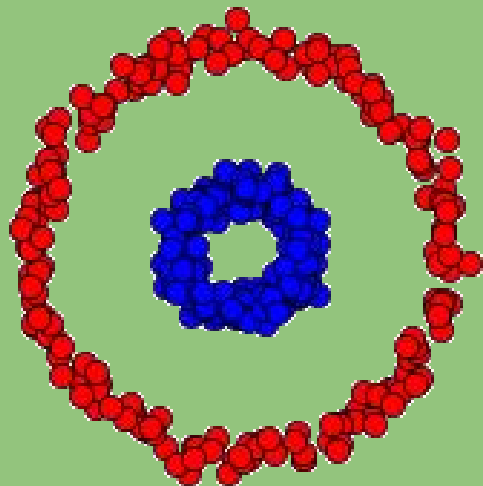
FEATURE TRANSFORMATION.

- Hmm... still linear.



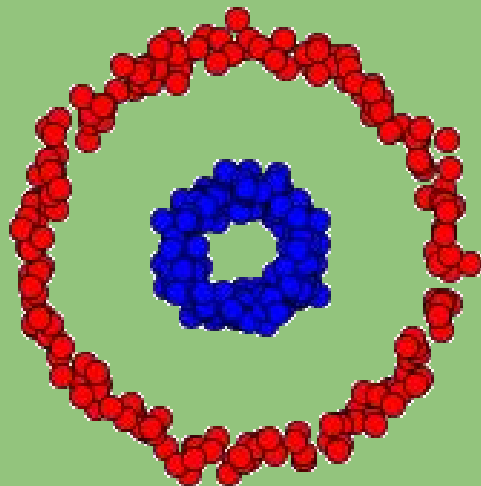
FEATURE TRANSFORMATION.

- How about...



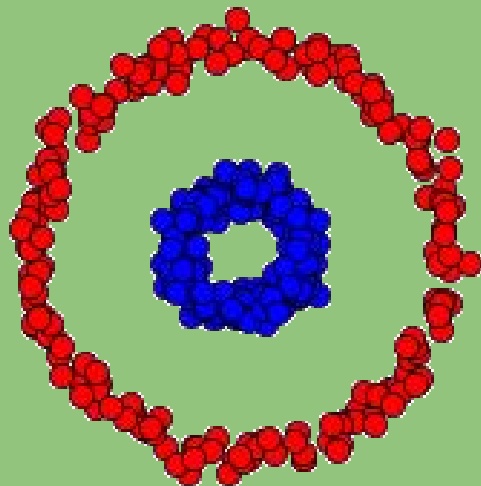
FEATURE TRANSFORMATION.

- How about...
- How do we transform the (x, y) features here?



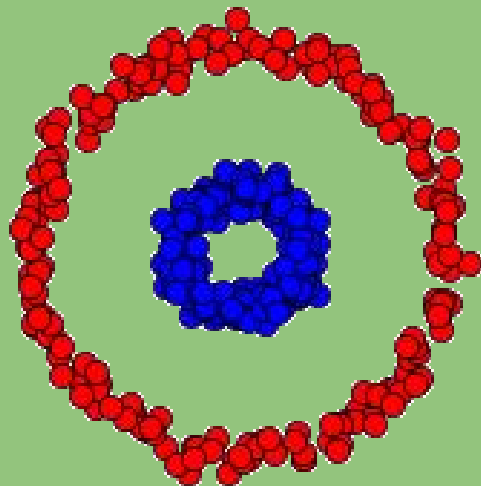
FEATURE TRANSFORMATION.

- How about...
- How do we transform the (x, y) features here?
 - $x \rightarrow x^2$
 - $y \rightarrow y^2$



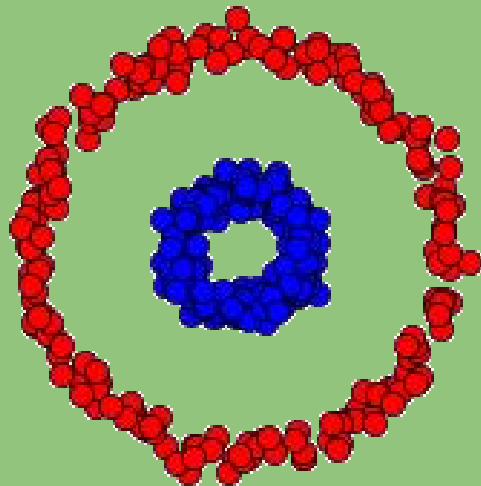
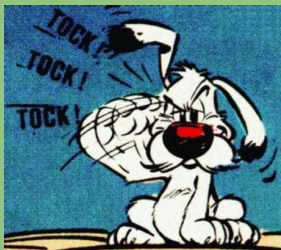
FEATURE TRANSFORMATION.

- How about...
- How do we transform the (x, y) features here?
 - $x \rightarrow x^2$
 - $y \rightarrow y^2$
- Uh, but this is *drawkcab*!
 - We visualized the data-set
 - Then figured out the right feature transformation!



FEATURE TRANSFORMATION.

- How about...
- How do we transform the (x, y) features here?
 - $x \rightarrow x^2$
 - $y \rightarrow y^2$
- Uh, but this is *drawkcab*!
 - We visualized the data-set
 - Then figured out the right feature transformation!



TAKE-AWAY

- Linear classifiers are stronger than they seem.
- Feature engineering is necessary at times.

TAKE-AWAY

- Linear classifiers are stronger than they seem.
- Feature engineering is necessary at times.

Hmm.. How do we evaluate classifiers though?

EVALUATION OF MODELS

How do we evaluate a trained model?

- Precision
- Recall
- Accuracy
- ...

EVALUATION OF MODELS

- The confusion matrix

		Predicted label	
		Y	N
Actual label	Y	34	7
	N	9	45

EVALUATION OF MODELS

- The confusion matrix

- Accuracy = (TP + TN)/ALL
 - = 79/95 = 83%

		Predicted label	
		Y	N
Actual label	Y	(TP) 34	(FN) 7
	N	(FP) 9	(TN) 45

EVALUATION OF MODELS

- The confusion matrix

		Predicted label	
		Y	N
Actual label	Y	(TP) 34	(FN) 7
	N	(FP) 9	(TN) 45

- Precision = $TP / (TP + FP)$
 - = $34 / 43 = 79\%$

EVALUATION OF MODELS

- The confusion matrix

		Predicted label	
		Y	N
Actual label	Y	(TP) 34	(FN) 7
	N	(FP) 9	(TN) 45

- $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$
 - $= 34 / 41 = 82\%$

EVALUATION OF MODELS

- The confusion matrix

		Predicted label	
		Y	N
Actual label	Y	(TP) 34	(FN) 7
	N	(FP) 9	(TN) 45

- Accuracy = $(TP + TN)/ALL$
 - = $79/95 = 83\%$
- Precision = $TP/(TP + FP)$
 - = $34/43 = 79\%$
- Recall = $TP/(TP + FN)$
 - = $34/41 = 82\%$

EVALUATION OF MODELS: RIDDLE

- The confusion matrix

		Predicted label	
		Y	N
Actual label	Y	(TP) ?	(FN) ?
	N	(FP) ?	(TN) ?

- $\text{Accuracy} = (\text{TP} + \text{TN}) / \text{ALL}$
- $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$
- $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$

Design a confusion matrix so that

- Accuracy is high
- Precision is low
- Recall is low

SIDEWALK 1: HANDS-ON LOGISTIC REGRESSION

- We will conjure up some random data
 - Then run logistic regression on this
-
- Tools used: sklearn, numpy.
 - \$ ipython notebook

SIDEWALK 2: SENTIMENT ANALYSIS

What is sentiment analysis?

- From wikipedia, *“to extract subjective information in source materials”*.
- Why is it useful?
 - Also called opinion mining.
 - Marketing Research
 - Customer feedback
 - Ratings for movies, hotels, books etc.

SIDEWALK 2: SENTIMENT ANALYSIS

What is sentiment analysis?

- From wikipedia, *“to extract subjective information in source materials”*.
- Quiz - who said what?
 - “This is not a novel to be tossed aside lightly. It should be thrown with great force”
 - “ From the moment I picked your book up until I laid it down, I was convulsed with laughter. Someday I intend reading it.”
 - “When I was a kid, my parents moved a lot, but I always found them.”
 - “I watched this play at a disadvantage. The curtain was up.”

SIDEWALK 2: SENTIMENT ANALYSIS

Sentimental people:



TASK: TO BUILD A SENTIMENT ANALYZER

We need:

- Labeled data!
 - Labels are “Positive”, “Negative”, and “Neutral”.
 - How much data?
- Partition labeled data into
 - Training
 - Test
- We will use IMDb movie reviews, and test it against new movies:
 - Dr. Strange
 - Arrival

TASK: TO BUILD A SENTIMENT ANALYZER

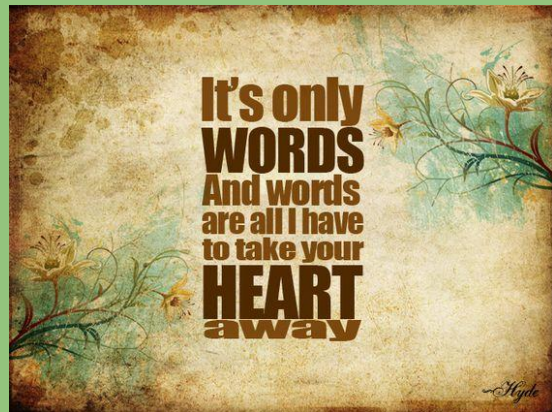
We need:

- Classifier of choice:
 - Naive Bayes. Based on Bayes' theorem.
- Metrics:
 - Precision, Recall, Accuracy.

TASK: TO BUILD A SENTIMENT ANALYZER

We need:

- Features?
 - Words!
 - And combinations
 - BIGRAMS
 - TRIGRAMS
 - ORTHOGONAL SPARSE BIGRAMS, ETC.



WHAT IS... BAYES' THEOREM?

Bayes Theorem relates a conditional probability and its reverse.

Wait, wha....t?

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}.$$

FIRST MIND WARP - WHY PROBABILITIES?!!

Most of the classifiers that we will see today are what are called

- Probabilistic classifiers

FIRST MIND WARP - WHY PROBABILITIES?!?

Most of the classifiers that we will see today are what are called

- Probabilistic classifiers
- I.e. they do not just tell us if an item belongs to this class or that, but instead gives a score (between 0 and 1)!

FIRST MIND WARP - WHY PROBABILITIES?!!

Most of the classifiers that we will see today are what are called

- Probabilistic classifiers
- I.e. they do not just tell us if an item belongs to this class or that, but instead gives a score (between 0 and 1)!
- Think of these as “probabilities” of belonging to one class or the other.

FIRST MIND WARP - WHY PROBABILITIES?!!

Most of the classifiers that we will see today are what are called

- Probabilistic classifiers
- I.e. they do not just tell us if an item belongs to this class or that, but instead gives a score (between 0 and 1)!
- Think of these as “probabilities” of belonging to one class or the other.
- Overkill?

FIRST MIND WARP - WHY PROBABILITIES?!?

Most of the classifiers that we will see today are what are called

- Probabilistic classifiers
- I.e. they do not just tell us if an item belongs to this class or that, but instead gives a score (between 0 and 1)!
- Think of these as “probabilities” of belonging to one class or the other.
- Overkill?
 - Not at all – simply good design.
 - Lots of times this is only an upstream scoring that goes into downstream decision making.

IDEA BEHIND NAIVE BAYES

Essential question for classification:

- Given a sentence “S” is it positive or negative (forget about neutral for now)?

IDEA BEHIND NAIVE BAYES

Essential question for classification:

- Given a sentence “S” is it positive or negative (forget about neutral for now)?
- I.e. in terms of probability, we are trying to score
 - $P(\text{“positive”} \mid S)$ and also hence
 - $P(\text{“negative”} \mid S)$

IDEA BEHIND NAIVE BAYES

Essential question for classification:

- Given a sentence “S” is it positive or negative (forget about neutral for now)?
- I.e. in terms of probability, we are trying to score
 - $P(\text{“positive”} \mid S)$ and also hence
 - $P(\text{“negative”} \mid S)$
- What does Bayes’ theorem do for us?

IDEA BEHIND NAIVE BAYES

Essential question for classification:

- Given a sentence “S” is it positive or negative (forget about neutral for now)?
- I.e. in terms of probability, we are trying to score
 - $P(\text{“positive”} \mid S)$ and also hence
 - $P(\text{“negative”} \mid S)$
- What does Bayes’ theorem do for us?
- Connects this up with $P(S \mid \text{“positive”})$, $P(S \mid \text{“negative”})$!

IDEA BEHIND NAIVE BAYES

Repeat in English

- Connects this up with $P(S \mid \text{"positive"})$, $P(S \mid \text{"negative"})$!
- Consider all the “positive” sentences in the labeled data, how often do we expect to “find something like” S in that data?

IDEA BEHIND NAIVE BAYES

Repeat in English

- Connects this up with $P(S \mid \text{"positive"})$, $P(S \mid \text{"negative"})$!
- Consider all the “positive” sentences in the labeled data, how often do we expect to “find something like” S in that data?
- What does “finding something like” S mean?

IDEA BEHIND NAIVE BAYES

Repeat in English

- Connects this up with $P(S \mid \text{“positive”})$, $P(S \mid \text{“negative”})$!
- Consider all the “positive” sentences in the labeled data, how often do we expect to “find something like” S in that data?
- What does “finding something like” S mean?
 - If we are trying to look for something pretty close to S , in the labeled data, then chances are we will not find it. People are expressive, there are too many ways to describe something good (or bad).
 - Called the “sparsity problem”.
 - So we should look at the “ingredients” that go to make up S instead!

IDEA BEHIND NAIVE BAYES

Repeat in English

- Connects this up with $P(S \mid \text{“positive”})$, $P(S \mid \text{“negative”})$!
- Consider all the “positive” sentences in the labeled data, how often do we expect to “find something like” S in that data?
- What does “finding something like” S mean?
 - Well, really, the words in S .
 - Say, if $S = \text{“This bike is good”}$. How often do we expect to see the word “good” among the positively labeled sentences in our training data?

IDEA BEHIND NAIVE BAYES

Other details:

- (conditional) independence assumption
 - The words in a document/sentence are “conditionally” independent given the class.
 - Example?
- This takes care of
 - The sparsity problem
 - Breaks up the computation of probabilities into manageable pieces.

IDEA BEHIND NAIVE BAYES

Other details:

- (conditional) independence assumption
 - The words in a document/sentence are “conditionally” independent given the class.
 - Example?
- This takes care of
 - The sparsity problem
 - Breaks up the computation of probabilities into manageable pieces.
- However, new problem arises:
 - What if we haven't seen a word at all? How do we manage that?

IDEA BEHIND NAIVE BAYES

Other details:

- However, new problem arises:
 - What if we haven't seen a word at all? How do we manage that?

IDEA BEHIND NAIVE BAYES

Other details:

- However, new problem arises:
 - What if we haven't seen a word at all? How do we manage that?
- We don't allow these conditional probabilities to vanish to zero.
 - Essentially jerky behavior:
 - **IF WORD PRESENT, THEN NON-ZERO CONDITIONAL PROBABILITY**
 - **IF WORD NOT SEEN AS YET, JUMPS TO ZERO CONDITIONAL PROBABILITY.**
 - Apply *smoothing*.

ENOUGH! SHOW ME THE CODE!

ENOUGH! SHOW ME THE CODE!



ENOUGH! SHOW ME THE CODE!

- import nltk.

```
10 """
11 A SentimentAnalyzer is a tool to implement and facilitate Sentiment Analysis tasks
12 using NLTK features and classifiers, especially for teaching and demonstrative
13 purposes.
14 """
15
16 from __future__ import print_function
17 from collections import defaultdict
18
19 from nltk.classify.util import apply_features, accuracy as eval_accuracy
20 from nltk.collocations import BigramCollocationFinder
21 from nltk.metrics import (BigramAssocMeasures, precision as eval_precision,
22                           recall as eval_recall, f_measure as eval_f_measure)
23
24 from nltk.probability import FreqDist
25
26 from nltk.sentiment.util import save_file, timer
27
28 class SentimentAnalyzer(object):
29     """
30     A Sentiment Analysis tool based on machine learning approaches.
31     """
```

SIDEWALK 3: DEEP LEARNING

Recall the issues with feature engineering.

- We needed to know the data well, to know the data well (i.e. to classify it).
- What if we were able to “learn” the (at times non-linear) features to be used?

SIDEWALK 3: DEEP LEARNING

Recall the issues with feature engineering.

- We needed to know the data well, to know the data well (i.e. to classify it).
- What if we were able to “learn” the (at times non-linear) features to be used?
- That is precisely what Deep Learning tries to do.

SIDEWALK 3: DEEP LEARNING

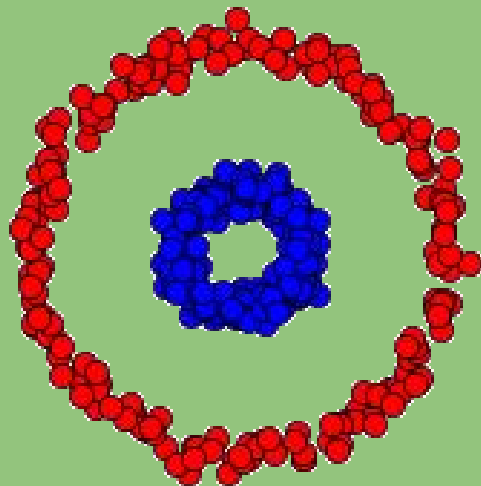
- That is precisely what Deep Learning tries to do.
- Stacks classifiers on top of classifiers.
- Outputs of earlier “coarser” classifiers are non-linear inputs to the latter layers.

SIDEWALK 3: DEEP LEARNING

- That is precisely what Deep Learning tries to do.
- Stacks classifiers on top of classifiers.
- Outputs of earlier “coarser” classifiers are non-linear inputs to the latter layers.
- Look at the earlier example again.

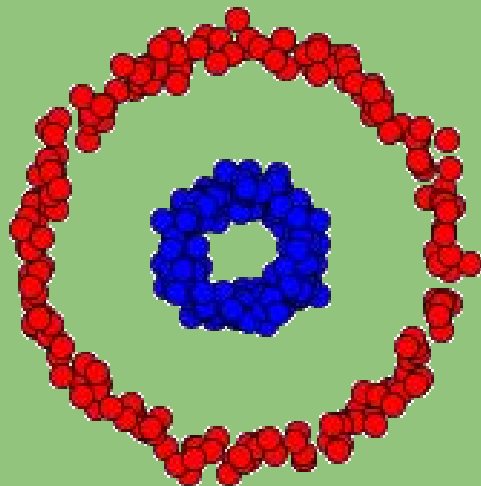
SIDEWALK 3: DEEP LEARNING

- That is precisely what Deep Learning tries to do.
- Stacks classifiers on top of classifiers.
- Outputs of earlier “coarser” classifiers are non-linear inputs to the latter layers.
- Look at the earlier example again.



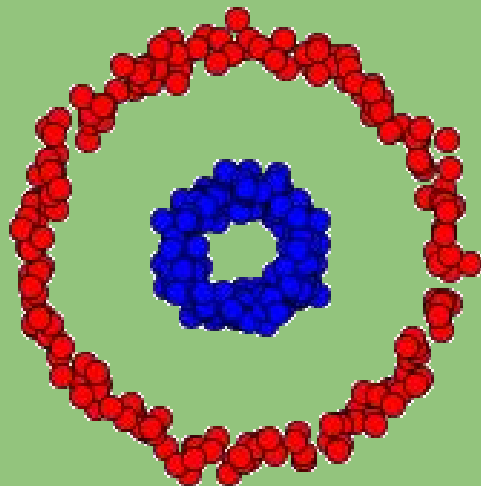
SIDEWALK 3: DEEP LEARNING

- Look at the earlier example again.
- Suppose we gave the machine features of the form
 - x^c without specifying c
- Would the machine be able to figure out for itself that $c = 2$?



SIDEWALK 3: DEEP LEARNING

- Look at the earlier example again.
 - Suppose we gave the machine features of the form
 - x^c without specifying c
 - Would the machine be able to figure out for itself that $c = 2$?
-
- Perhaps...
 - Given enough data.
 - Smarter training procedures.



SIDEWALK 3: DEEP LEARNING.

How does this tie up with the classifier-on-classifier story?

Well,

- a first level classifier is trying to figure out the “correct” feature transform $x \rightarrow x^2$, and
- the second level classifier is trying to run a line through the (thereby) modified data.

SIDEWALK 3: DEEP LEARNING.

How does this tie up with the classifier-on-classifier story?

Well,

- a first level classifier is trying to figure out the “correct” feature transform $x \rightarrow x^2$, and
- the second level classifier is trying to run a line through the (thereby) modified data.

Surprisingly simple, and stunningly powerful idea!

WELL... ALL IS HUNKYDORY BUT...

- Note now that instead of the handcrafted x^2 , we have a free parameter c in x^c .

WELL... ALL IS HUNKYDORY BUT...

- Note now that instead of the handcrafted x^2 , we have a free parameter c in x^c .
- A lot more parameters in a deep network!
 - Need to find these parameters – this is what “learning” amounts to.

WELL... ALL IS HUNKYDORY BUT...

- A lot more parameters in a deep network!
 - Need to find these parameters – this is what “learning” amounts to.

WELL... ALL IS HUNKYDORY BUT...

- A lot more parameters in a deep network!
 - Need to find these parameters – this is what “learning” amounts to.
- Insight from linear equations:
 - Underdetermined systems. Fewer equations/constraints than variables are a problem. Cannot have unique solutions.
 - Think of each training example as a “constraint” (constraining the network to output something close to the “label” on that training input).

WELL... ALL IS HUNKYDORY BUT...

- A lot more parameters in a deep network!
 - Need to find these parameters – this is what “learning” amounts to.
- Insight from linear equations:
 - Underdetermined systems. Fewer equations/constraints than variables are a problem. Cannot have unique solutions.
 - Think of each training example as a “constraint” (constraining the network to output something close to the “label” on that training input).
- More the number of parameters, more the number of training/labeled examples required.
 - In the age of big data, plenty of unlabeled data.

SHALLOW VS. DEEP LEARNING

- A shallow classifier (eg. SVM, Log Reg) is trying to learn a function
 - $y = F(x)$
- OTOH, a deep classifier (say of depth 3, 2 hidden layers) is trying to learn a function
 - $y = F \circ G \circ H(x)$, where “o” = composition.

SHALLOW VS. DEEP LEARNING - WHAT GIVES?

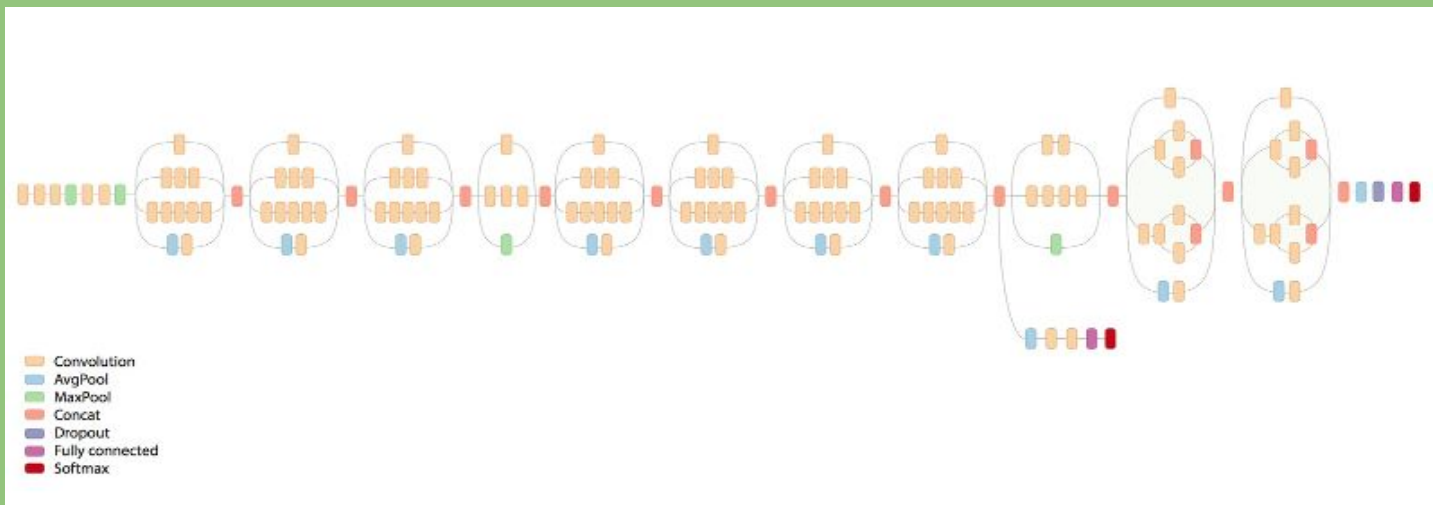
- But $F \circ G \circ H$ is just yet another function!
- ???

CNN - AND NOW THE NEWS...



DEEP LEARNING MODELS FOR IMAGES.

- Recent fascinating work has used deep networks (involving what are called “convolutions”) to classify images.
- Used a large publicly available training data set, called ImageNet.
- Error has dropped to ~3%!



DEEP LEARNING MODELS FOR LANGUAGE.

- Recent fascinating work has resurrected recurrent neural nets, LSTMs for
 - Machine translation
 - Sentiment analysis
 - Language modeling
 - Many many others.

ENDGAME

- Please fill up survey with comments.
- We will feed it to this very sentiment analyzer and come up with an aggregate score!

ENDGAME

- Please fill up survey with comments.
- We will feed it to this very sentiment analyzer and come up with an aggregate score!

Just Kidding!

Thank You!!!

ENDGAME

Thank You!!!

EVALUATION OF MODELS

- The confusion matrix

		Predicted label	
		Y	N
Actual label	Y	34	7
	N	9	45